



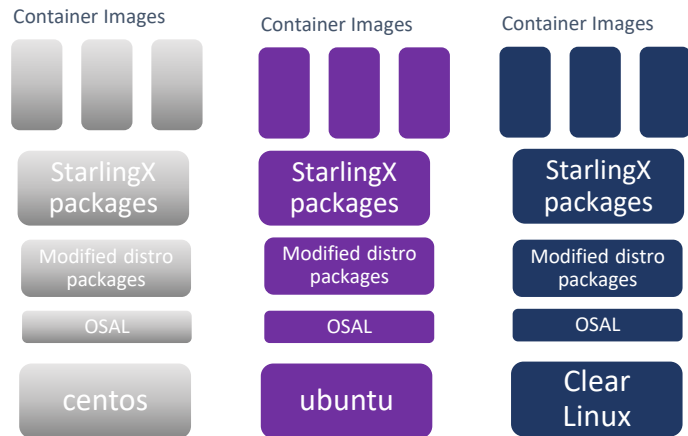
# Multi-OS Discussion

# Objective

- Support StarlingX on multiple OS distro's providing **equivalent** functionality on each
- Initial Candidate OS's: Centos (current), Ubuntu, Clear Linux

# Overview

- Starting point is vanilla OS distribution packages required packages for StarlingX
- Modified OS packages built against the distro
  - Goal is to drive this to virtual zero
- StarlingX packages are built against each distro
- Create a OS abstraction layer - OSAL
  - Bridge between the common code and OS to translate the requests to OS dependent function
  - This will be needed for subsystems such as patching
- Containerize the infrastructure to provide an OS agnostic packaging format
  - Ideally everything outside of the base OS would move to containers
  - Potentially a set of images built against one distro can be used for all



## Current OS Distro

- StarlingX is currently not based on a pure centos distribution:
  - Skinny distribution, minimum set of packages
  - Number of modified distro packages (100-150)
    - Need to drive this to virtual zero
  - Packages that are built out of tree to pick up newer drivers, qemu etc.
  - Some required packages are not even provided by Centos distro

## Key Building Block Alignment

- In order to ensure maximum feature alignment and minimize validation costs, it is recommended to align on versions of key packages:
  - python
  - kubernetes & friends
  - libvirt/QEMU
  - networking drivers
  - OVS-DPDK
  - CEPH
  - Openstack version
  - Others ?
- These would need to be built out of tree against each distro

## Other Considerations

- All OS's will not provide equivalent functionality
  - For instance Ubuntu does not provide a RT kernel, therefore the current low latency compute profile does not make sense for it

# Build Environment

- Need a build environment to build packages against each supported distro
- Need to be able to build patched packages for each distro
  - Patches will likely be different as the package versions will vary
- Build environment needs to lock down package versions versus letting float
  - At the very least this is required for release loads

# Package Customizations

- Need to reduce the number of customizations – upstreaming, refactoring, deprecation
- For the remainder
  - Refactor patches for each distro and provide custom per distro packages
  - Very expensive if this ends up being a big number



# Containerizing the infrastructure

- Containerize the infrastructure to provide a consistent packaging format across all distro's
  - Containerizing openstack is the initial step
- This avoids having per distro package management and s/w live cycle implementations
- Build once and use across all ?
  - Container images built against one distro can be used for all

# Package Management

- Centos uses rpm with yum or smart as the package manager
- Ubuntu uses deb with apt as the package manager
- Software management functions such as patching will need to be abstracted away from the underlying package management
  - Patching would need to be updated to abstract away the package manager
    - Interfaces with the correct package manager based on distro type

# Installer

- Each distro will have it's own installer that will need to be integrated into StarlingX
  - Anaconda for centos which has been extended by StarlingX
  - Kickstart equivalents for other installers would need to be updated

# Initialization

- All considered distro's use systemd